Summary

ORMs, what is the difference between Entity Frameworks and Dapper?

**ORM (object-relational mapping):** programming technique to convert data from relational databases to objects. This is typically needed because objects are generally non-scalar and database tables are scalar.

ORM advantage: can often reduce the amount of code that needs to be written.

ORM disadvantage: high level of abstraction.

* Some argue this has resulted in an increase in poorly designed databases…

Entity Framework:

Dapper:

Details:

1. What are the pros/cons of both. (**why would you choose one over the other**).
   1. Performance
   2. Ease of setup
   3. Ease of use from a developer’s point of view. (small and large projects/systems)
   4. Transactions
   5. Current trend is developing micro services (which one lends it’s self to that type of architecture)
2. What do the DTOs do verses a domain object.
3. Write an example of how both handle CRUD operations with a SQL server.

<https://dapper-tutorial.net/dapper>

<https://docs.microsoft.com/en-us/ef/core/>

Questions/thoughts?

Sources:

<https://en.wikipedia.org/wiki/Object-relational_mapping>

<https://en.wikipedia.org/wiki/Entity_Framework>

<https://en.wikipedia.org/wiki/Dapper_ORM>

<https://dapper-tutorial.net/dapper>

<https://docs.microsoft.com/en-us/ef/core/>